Network Working Group
Request for Comments: 4168
Category: Standards Track

J. Rosenberg
Cisco Systems
H. Schulzrinne
Columbia University
G. Camarillo
Ericsson

October 2005

The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)

Status of This Memo

Copyright Notice

Abstract

This document specifies a mechanism for usage of SCTP (the Stream Control Transmission Protocol) as the transport mechanism between SIP (Session Initiation Protocol) entities. SCTP is a new protocol that provides several features that may prove beneficial for transport between SIP entities that exchange a large amount of messages, including gateways and proxies. As SIP is transport-independent, support of SCTP is a relatively straightforward process, nearly identical to support for TCP.

Table of Contents

1. Introduction

The Stream Control Transmission Protocol (SCTP) [4] has been designed as a new transport protocol for the Internet (or intranets) at the same layer as TCP and UDP. SCTP has been designed with the transport of legacy SS7 signaling messages in mind. We have observed that many of the features designed to support transport of such signaling are also useful for the transport of SIP (the Session Initiation Protocol) [5], which is used to initiate and manage interactive sessions on the Internet.

SIP itself is transport-independent, and can run over any reliable or unreliable message or stream transport. However, procedures are only defined for transport over UDP and TCP. This document defines transport of SIP over SCTP.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

3. Potential Benefits

RFC 3257 presents some of the key benefits of SCTP [10]. We summarize some of these benefits here and analyze how they relate to SIP (a more detailed analysis can be found in [12]).

3.1. Advantages over UDP

All the advantages that SCTP has over UDP regarding SIP transport are also shared by TCP. Below, there is a list of the general advantages that a connection-oriented transport protocol such as TCP or SCTP has over a connection-less transport protocol such as UDP.

Fast Retransmit: SCTP can quickly determine the loss of a packet, because of its usage of SACK and a mechanism that sends SACK messages faster than normal when losses are detected. The result is that losses of SIP messages can be detected much faster than when SIP is run over UDP (detection will take at least 500 ms, if not more). Note that TCP SACK exists as well, and TCP also has a fast retransmit option. Over an existing connection, this results in faster call setup times under conditions of packet loss, which is very desirable. This is probably the most significant advantage of SCTP for SIP transport.

Congestion Control: SCTP maintains congestion control over the entire association. For SIP, this means that the aggregate rate of messages between two entities can be controlled. When SIP is run over TCP, the same advantages are afforded. However, when run over UDP, SIP provides less effective congestion control. This is because congestion state (measured in terms of the UDP retransmit interval) is computed on a transaction-by-transaction basis, rather than across all transactions. Thus, congestion control performance is similar to opening N parallel TCP connections, as opposed to sending N messages over one TCP connection.

Transport-Layer Fragmentation: SCTP and TCP provide transport-layer fragmentation. If a SIP message is larger than the MTU size, it is fragmented at the transport layer. When UDP is used, fragmentation occurs at the IP layer. IP fragmentation increases the likelihood of having packet losses and makes NAT and firewall traversal difficult, if not impossible. This feature will become important if the size of SIP messages grows dramatically.

3.2. Advantages over TCP

We have shown the advantages of SCTP and TCP over UDP. We now analyze the advantages of SCTP over TCP.

Head of the Line: SCTP is message-based, as opposed to TCP, which is stream-based. This allows SCTP to separate different signalling messages at the transport layer. TCP only understands bytes. Assembling received bytes to form signalling messages is performed at the application layer. Therefore, TCP always delivers an

ordered stream of bytes to the application. On the other hand, SCTP can deliver signalling messages to the application as soon as they arrive (when using the unordered service). The loss of a signalling message does not affect the delivery of the rest of the messages. This avoids the head of line blocking problem in TCP, which occurs when multiple higher layer connections are multiplexed within a single TCP connection. A SIP transaction can be considered an application layer connection. There are multiple transactions running between proxies. The loss of a message in one transaction should not adversely effect the ability of a different transaction to send a message. Thus, if SIP is run between entities with many transactions occurring in parallel, SCTP can provide improved performance over SIP over TCP (but not SIP over UDP; SIP over UDP is not ideal from a congestion control standpoint; see above).

Easier Parsing: Another advantage of message-based protocols, such as SCTP and UDP, over stream-based protocols, such as TCP, is that they allow easier parsing of messages at the application layer. There is no need to establish boundaries (typically using Content-Length headers) between different messages. However, this advantage is almost negligible.

Multihoming: An SCTP connection can be associated with multiple IP addresses on the same host. Data is always sent over one of the addresses, but if it becomes unreachable, data sent to one can migrate to a different address. This improves fault tolerance; network failures making one interface of the server unavailable do not prevent the service from continuing to operate. SIP servers are likely to have substantial fault tolerance requirements. It is worth noting that, because SIP is message oriented and not stream oriented, the existing SRV (Service Selection) procedures defined in [5] can accomplish the same goal, even when SIP is run over TCP. In fact, SRV records allow the 'connection' to fail over to a separate host. Since SIP proxies can run statelessly, failover can be accomplished without data synchronization between the primary and its backups. Thus, the multihoming capabilities of SCTP provide marginal benefits.

It is important to note that most of the benefits of SCTP for SIP occur under loss conditions. Therefore, under a zero loss condition, SCTP transport of SIP should perform on par with TCP transport. Research is needed to evaluate under what loss conditions the improvements in setup times and throughput will be observed.

4. Transport Parameter

Via header fields carry a transport protocol identifier. RFC 3261 defines the value "SCTP" for SCTP, but does not define the value for the transport parameter for TLS over SCTP. Note that the value "TLS", defined by RFC 3261, is intended for TLS over TCP.

Here we define the value "TLS-SCTP" for the transport part of the Via header field to be used for requests sent over TLS over SCTP [8]. The updated augmented BNF (Backus-Naur Form) [2] for this parameter is the following (the original BNF for this parameter can be found in RFC 3261):

```
transport          =  "UDP" / "TCP" / "TLS" / "SCTP" / "TLS-SCTP"
                      / other-transport
```

The following are examples of Via header fields using "SCTP" and "TLS-SCTP":

```
Via: SIP/2.0/SCTP ws1234.example.com:5060
Via: SIP/2.0/TLS-SCTP ws1234.example.com:5060
```

5. SCTP Usage

Rules for sending a request over SCTP are identical to TCP. The only difference is that an SCTP sender has to choose a particular stream within an association in order to send the request (see Section 5.1).

Note that no SCTP identifier needs to be defined for SIP messages. Therefore, the Payload Protocol Identifier in SCTP DATA chunks transporting SIP messages MUST be set to zero.

The SIP transport layers of both peers are responsible for managing the persistent SCTP connection between them. On the sender side, the core or a client (or server) transaction generates a request (or response) and passes it to the transport layer. The transport sends the request to the peer’s transaction layer. The peer’s transaction layer is responsible for delivering the incoming request (or response) to the proper existing server (or client) transaction. If no server (or client) transaction exists for the incoming message, the transport layer passes the request (or response) to the core, which may decide to construct a new server (or client) transaction.

5.1. Mapping of SIP Transactions into SCTP Streams

SIP transactions need to be mapped into SCTP streams in a way that avoids Head Of the Line (HOL) blocking. Among the different ways of performing this mapping that fulfill this requirement, we have chosen

the simplest one; a SIP entity SHOULD send every SIP message (request or response) over stream zero with the unordered flag set. On the receiving side, a SIP entity MUST be ready to receive SIP messages over any stream.

> In the past, it was proposed that SCTP stream IDs be used as lightweight SIP transaction identifiers. That proposal was withdrawn because SIP now provides (as defined in RFC 3261 [5]) a transaction identifier in the branch parameter of the Via entries. This transaction identifier, missing in the previous SIP spec [9], makes it unnecessary to use the SCTP stream IDs to demultiplex SIP traffic.

In many circumstances, SIP requires the use of TLS [3], for instance, when routing a SIPS URI [5]. As defined in RFC 3436 [8], TLS running over SCTP MUST NOT use the SCTP unordered delivery service. Moreover, any SIP use of an extra layer between the transport layer and SIP that requires ordered delivery of messages MUST NOT use the SCTP unordered delivery service.

SIP applications that require ordered delivery of messages from the transport layer (e.g., TLS) SHOULD send SIP messages belonging to the same SIP transaction over the same SCTP stream. Additionally, they SHOULD send messages belonging to different SIP transactions over different SCTP streams, as long as there are enough available streams.

> A common scenario where the above mechanism should be used consists of two proxies exchanging SIP traffic over a TLS connection using SCTP as the transport protocol. This works because all of the SIP transactions between the two proxies can be established within one SCTP association.

Note that if both sides of the association follow this recommendation, when a request arrives over a particular stream, the server is free to return responses over a different stream. This way, both sides manage the available streams in the sending direction, independently of the streams chosen by the other side to send a particular SIP message. This avoids undesirable collisions when seizing a particular stream.

6. Locating a SIP Server

The primary issue when sending a request is determining whether the next hop server supports SCTP so that an association can be opened. SIP entities follow normal SIP procedures to discover [6] a server that supports SCTP.

However, in order to use TLS on top of SCTP, an extra definition is needed. RFC 3263 defines the NAPTR (Naming Authority Pointer) [7] service value "SIP+D2S" for SCTP, but fails to define a value for TLS over SCTP. Here we define the NAPTR service value "SIPS+D2S" for servers that support TLS over SCTP [8].

7. Security Considerations

The security issues raised in RFC 3261 [5] are not worsened by SCTP, provided the advice in Section 5.1 is followed and TLS over SCTP [8] is used where TLS would be required in RFC 3261 [5] or in RFC 3263 [6]. So, the mechanisms described in RFC 3436 [8] MUST be used when SIP runs on top of TLS [3] and SCTP.

8. IANA Considerations

This document defines a new NAPTR service field value (SIPS+ D2S). The IANA has registered this value under the "Registry for the SIP SRV Resource Record Services Field". The resulting entry is as follows:

Services Field	Protocol	Reference
SIPS+D2S	SCTP	[RFC4168]

9. References

9.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[3] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[4] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[6] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[7] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[8] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol", RFC 3436, December 2002.

9.2. Informative References

[9] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[10] Coene, L., "Stream Control Transmission Protocol Applicability Statement", RFC 3257, April 2002.

[11] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)", BCP 99, RFC 3969, December 2004.

[12] Camarillo, G., Schulrinne, H., and R. Kantola, "Evaluation of Transport Protocols for the Session Initiation Protocol", IEEE, Network vol. 17, no. 5, 2003.

Authors' Addresses

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
EMail: jdrosen@cisco.com
URI: http://www.jdrosen.net

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003
US

EMail: schulzrinne@cs.columbia.edu

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.